Trinocular Stereo Vision for Intelligent Robot Navigation

Andersen, Jens Christian; Andersen, Nils Axel; Ravn, Ole

Published in:
Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles

Publication date:
2004

TRINOCULAR STEREO VISION FOR INTELLIGENT ROBOT NAVIGATION

Jens Christian Andersen * **Nils A. Andersen** *
Ole Ravn *

* *Automation Ørsted•DTU, Technical University of Denmark, building 326, DK-2800 Kgs. Lyngby, Denmark.*
(jca,naa,or)@oersted.dtu.dk

Abstract: This paper describes a vision sensor that extracts visible features of objects using a set of on-board robot cameras. The purpose of the sensor is to be able to classify the seen objects into abstract object types like tables, chairs, walls and doors using these features. The results show that the vision sensor is able to extract a filtered set of features suitable for the purpose. The method used is stereoscopic scanning followed by a filtering process in 3D space. The method is insensitive to the shape of the objects and the structure of the background.

Keywords: Vision based navigation, autonomous robot, recognition

1. INTRODUCTION

The research project behind this paper is focused on autonomous robot navigation in an indoor human environment, where the robot is able to find and recognize some basic types of objects. The robot sensors include camera vision as the primary sensor for this task.

Mobile robot navigation is the discipline of locating own position and to find a path to the destination. Finding a path involves in this case recognition and classification of observed objects, so that the destination can be described in terms of the classified objects, e.g. as "go to the third table in the first room to the left".

The stereoscopic vision sensor provides the majority of the data needed for object classification. The extracted data should allow for classification into classes like chair, table, wall, door or human, without prior knowledge of where to expect what type of objects.

Stereoscopic vision systems have been described in a number of papers over the resent years. Most of these systems extract linear features in the images from single cameras and then correlate these to get the 3D information like in (Loaïza *et al.*, 2001), (Loaïza *et al.*, 1999), (Kim and Nevatia, 1999) and (Shah and Aggarwal, 1997). In (Se *et al.*, 2001), the images are searched for scale invariant features — e.g. line ends — before correlation and 3D calculation.

Most of the stereoscopic camera solutions use three cameras achieve better immunity to false correlations. Image processing for stereo calculation is processing power intensive and only a few papers document image processing times of less than one second. For obstacle mapping (Murray and Little, 2000) expect 3 updates per second, and, for robot position estimating (Se *et al.*, 2001), two estimates per second were obtained from the camera images using a 700 MHz PC. Faster is better, but processing-times in excess of one second is acceptable when the resulting data is to be used for route planning and not directly in the motion control loop.

Fig. 1. An image like this is easily decoded by a human, but requires explicit processing for a robot.

This paper describes a method where images are scanned for 3D points first and the filtering is done subsequently in 3D space.

2. OBJECTIVES AND OVERVIEW

When looking at an image as in Figure 1 it is easy to see that it shows a chair, a golf-ball on the floor, and some closets in the background. If a robot is to get to the same conclusion, there is a fair bit of processing is to be performed, some of which is described in the following.

To find out that the nearest object is a chair, a number of tasks must be performed. The object must be separated from the rest of the image, and it must be analyzed for properties that indicate it is a chair. The properties could be that it has a surface, the seat, of an appropriate size (e.g. 40x40 cm) at an appropriate height (approximately 50 cm), and that it has an open structure below the surface to support it. These properties would make it distinguishable from other objects like a box or a table.

From one images set the visible features can be extracted. The visible features of a chair could be the front edge of the seat, the chair bag or parts of the support structure. The idea is that a set of such features is sufficient to perform the needed object classification.

This paper describes a vision sensor and the processing behind the feature extraction.

The sensor has three major parts, the feature filtering, the stereo scanning and the camera calibration. Each of these parts are described below.

3. FEATURE FILTERING

The data from the stereo correlation is 3D positions reconstructed from corresponding points

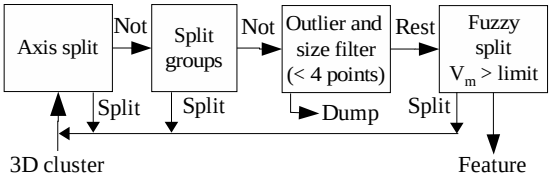

Fig. 2. The input clusters are groups of isolated 3D detections, these gets divided into features of just one object by splitting.

in a set of stereoscopic images. About 1000 3D positions are produced in a full stereoscopic image scanning. These positions are grouped into spatially separated clusters.

The feature filtering function must ensure that clusters represent one object only. As different objects often is seen close together — e.g. a table and a wall — they may not be separable on distance alone.

To separate these clusters into features from just one object, two primary methods are used. The first is to extract horizontal and vertically oriented features, and the second is to split combined features based on similar positional and color properties.

In an indoor environment, horizontal and vertical features are often found in especially empty rooms. These features are extracted first based a density histogram along the vertical axis for the horizontal lines. This will typically produce a peak in the histogram when the cluster contains a major feature perpendicular to the vertical axis. The peak is then isolated to a new cluster. The same method is used across the image, and this isolates primarily vertical lines.

A fuzzy c-means classifier from (Babuška, 1998) is able to separate 3D positions in a cluster of into two or more sub-clusters. The ellipsoid shape of the sub-clusters may differ, e.g. a cluster representing a chair seat in combination with one of its legs, may be divided into a narrow ellipsoid formed cluster from the leg and a more ball formed for the seat. The fuzzy c-means method needs to know the number of sub-clusters, and the computation afford increases rapidly with the number of clusters. A computational affordable method is to split clusters into two subclusters until the cluster statistics are reasonable, or the clusters are too small to split. This is the method used, and it produces slightly more clusters than is strictly necessary, as the only drawback.

The cluster statistics is calculated based on the best-fit 3D line. The 3D line is the major axis of the ellipsoid that best describes the cluster, i.e. where the sum of the squared distance from the line to the 3D positions is minimum. The line is found by finding the best fit line in two dimensions

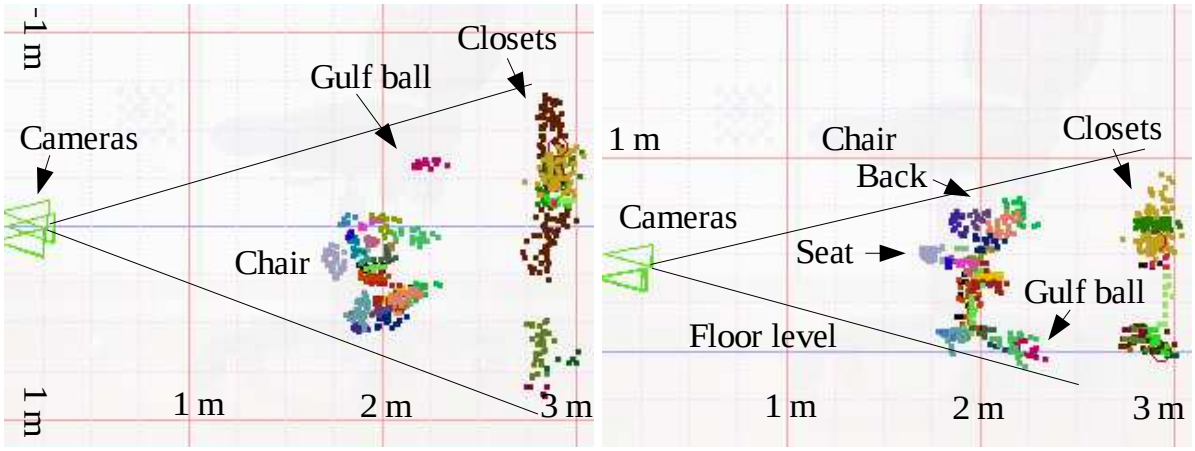

Fig. 3. The images show the features extracted from the scene in Figure 1. The left image is top view, and the right image is side view. The three camera positions are shown as triangles. Each feature is represented with a colored group of 3D detections.

twice. First in the horizontal plane and then in the 2D plane spanned by the vertical axis and one of the horizontal axes. The used axis combinations are selected so that singularities are best avoided.

A number of statistics are calculated along this 3D line with the endpoints $\mathbf{E}_1$ and $\mathbf{E}_2$. One of them is the mass distribution or inertial moment V_m for a unified line length, as shown in Equation 1.

$$V_m = \frac{\frac{1}{n}\sum_{j=1}^{n}\left(t(\mathbf{X}_j) - t(\mathbf{X}_g)\right)^2}{|\mathbf{E}_2 - \mathbf{E}_1|^2} \qquad (1)$$

Where $\mathbf{X}_g$ is the center of gravity and $t(\mathbf{X}_j)$ is the position on the line closest to the 3D point $\mathbf{X}_j$, all 3D points are given equal weight. For a uniform distribution along the 3D line, this value V_m would be $^1/_{12}$. Clusters are split if the mass distribution is above a value comparable with the uniform distribution.

Clusters are furthermore tested for outliers and for splits. Splits are clusters that are no longer continuous, i.e. there is an opening along the major cluster axis. Split clusters are divided into continuous subclusters.

The full feature extraction process is shown in Figure 2, and an example of the extracted features is shown in Figure 3.

The resulting clusters are assumed object features and are shown on top of one of the images in Figure 4.

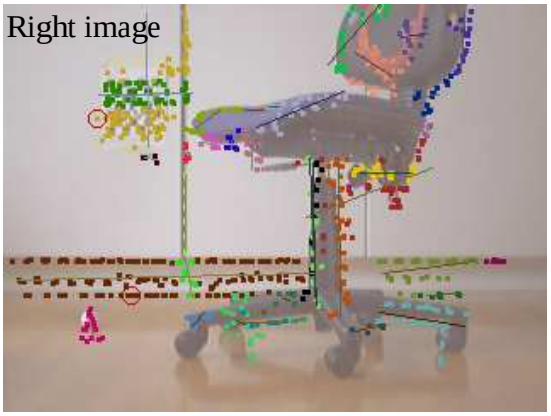

Fig. 4. The resulting clusters after filtering is shown on top of one of the original images. In total 848 3D detections are shown in 36 clusters.

4. STEREOSCOPIC SCANNING

The use of cameras to aid robot navigation has been described in a number of papers using a number of different approaches. A two camera solutions with fish-eye lenses is used for navigation in (Shah and Aggarwal, 1997) by finding significant lines ending in the geometric vanishing points, and from these and vertical lines build a model of the traversed corridor. Another project (Loaïza *et al.*, 1999) and (Loaïza *et al.*, 2001) uses two cameras mounted on top of each other and does stereoscopic calculation on contrast straight lines found in the images, the (Loaïza *et al.*, 2001) project puts emphasis in color on each side of the extracted lines to improve matching. The project (Se *et al.*, 2001) uses three cameras in a right angle triangular configuration to remove false correlations. The project also pre-extracts a number of (scale invariant) features before stereo calculation, and then stores these points (about 3000 for one room) for the ongoing navigation.

These projects all put a great deal of effort into extracting data from the two-dimensional images prior to the stereo calculation. This will reduce the stereo calculation effort, but objects are limited to the extracted types, typically linear features only. The method selected here does not extract features in the two-dimensional images but only

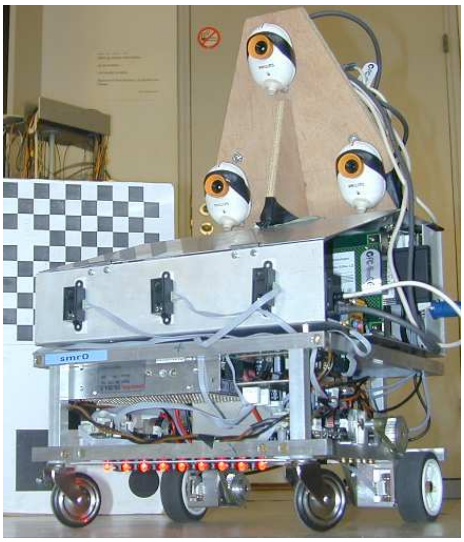

Fig. 5. The robot platform used for the experiments with the trinocular camera mount on top. The distance between any two cameras are 15 cm.

edges. This ensures that no object gets discarded based on its shape.

The camera configuration is as shown in Figure 5. The cameras are standard web cameras connected to the robot motherboard using a Universal Serial Bus (USB) interface. The robot main processor is using a Linux operating system (Redhat standard distribution).

The stereo scanning is performed using two camera images at a time. The reference image is divided into a number of scanning lines and for each of these the corresponding epipolar line is found in the other image. Both images are then resampled along these lines in 3 pixel wide bands. Contrast edges are found in the resampled reference image, and for each of these, a match is found in the other image. The cross-correlation coefficient ρ in range $[-1, 1]$ is found over a small mask area. The size of this correlation mask area depends of the resolution, and is 3x17 pixels at a 120x160 image resolution and 3x31 pixels for a 240x320 image resolution. The image is scanned in the part of the image that would result in a 3D distances from 0.5 meter to 20 meter from the cameras. To reduce processing the correlation is calculated for every pixel position is the last correlation was a close match — i.e. a ρ value above 0.2. For ρ below this value, from 1 to 6 pixels gets skipped before another correlation is attempted.

Figure 6 shows an example on a top and bottom-right image set. One of the edges on the floor line in the left image is found in the search area in the right image. The correlation coefficient is shown in the box in the right image, with the best correlation marked with a circle. The used image resolution is 320x240 pixels.

Each set of two images is scanned in this way, both with the first and the second image as the reference image. In total six scannings are thus performed for each set of three images.

The resultant set of 3D positions (about 1200 for the chair motif) are grouped into clusters, so that the 3D positions in a cluster are not separated by more than could be expected with the used epipolar line distance (9 pixels).

Each cluster should have 3D positions from at least four of the six scannings, otherwise the cluster is assumed false and dropped. This criterion removes almost all the false correlations. The remaining clusters are split and filtered as described above.

5. CAMERA CALIBRATION

The camera orientation is very important for a correct stereoscopic conversion. The used Web camera could not be mounted in a predetermined orientation with a reasonable accuracy, so the orientation must be determined after the cameras are mounted on the robot.

The general idea is that the camera *position* on the robot can be established reasonably accurate by measurement, whereas the *rotation* is difficult to establish by direct measurement. By placing a known calibration chart at some distance directly in front of the robot at a known height, it is possible to calculate a sufficient exact rotation around all three axes for each camera on the camera mount.

The calibration chart consists of a number of black squares on a white background, arranged as a checkerboard as shown behind the robot in Figure 5. The corner where two black squares touch is scale invariant and easily detected. Two of the black squares in the calibration chart are omitted and the position of these are used to determine the center of the calibration chart and from that the pixel position of all corners.

The pixel position $\mathbf{X}_p$ $[x_p, y_p]$ corresponding to each corner position $\mathbf{X}_c$ $[x_c, y_c, 0]$ on the flat calibration chart can be calculated as a function of chart position relative to the camera $[x_t, y_t, z_t]$ and the camera rotation $[\Omega, \Phi, \kappa]$ around the three axis. The pixel position $\mathbf{X}_p$ and the corresponding point $\mathbf{X}_c$ on the calibration chart are known for up to 100 positions. The camera position and rotation form this equation is then estimated using a least-square parameter adjustment method.

This method can determine the orientation of the cameras to about 0.01° for both the left-right orientation Φ and the up-down orientation Ω,

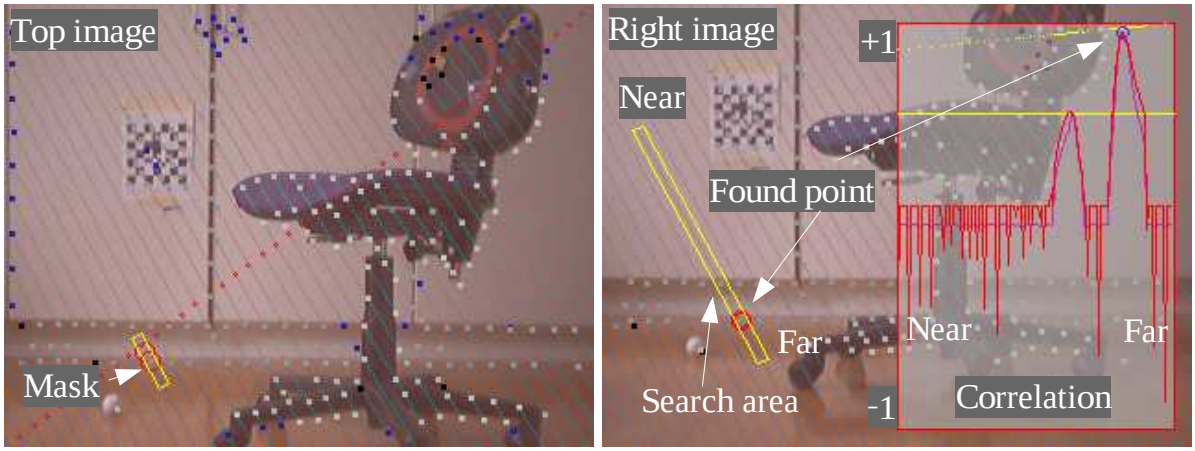

Fig. 6. The left and right image shows the epipolar lines used in the scanning process. The left image is the reference image (top camera) with one edge marked in the mask-sized box. The area marked in the right image is scanned for the same pattern. The correlation quotient is shown in the curve. The gray dots show the points where correlation is attempted — darker gray is further away.

and to about 0.03° for the rotation κ around the camera optical axis.

When using identical cameras with USB interfaces there is no guarantee that the cameras will be detected in the same order as last time they were plugged into the robot. The calibration method described can estimate the camera position to within 1–2 cm, which is sufficiently accurate to determine which camera device is at top, left and right position.

The used cameras (Philips PCVC 740K) have a focus length of about 1055 pixels with an image resolution of 640x480 pixels. The radial distortion is estimated to about $1.4 \cdot 10^{-7}$ proportional to r^3 and $8.5 \cdot 10^{-13}$ proportional with r^5, where r is radius from image center in pixels. These internal camera parameters are estimated off-line using a series of images of the calibration chart.

One image pixel corresponds to an angle of about 0.1° at a 320x240 resolution (The camera opening horizontally is 33.5°). A one-pixel error corresponds to about a 10 cm range error at 2 meter distance from the camera with the used camera configuration. Practical results shows an about 5 cm (standard deviation) distance error from the stereo calculation using a 320x240 image resolution.

6. RESULTS

The processing time from image capture to extracted features is shown in Table 1 for the three available camera resolutions.

The long capture time for 640x480 resolution is due to an interface limitation, where cameras have to be closed between images. All timings are from a 700 MHz PC and for a slightly simpler scene than in Figure 3.

	160x120	320x240	640x480	
Image delay	0.24	0.24	0.24	sec
Image capture	0.01	0.06	5.8	sec
Epipolar dist.	6	9	12	pix
Stereo scan.	0.53	1.71	4.35	sec
3D detections	487	744	1145	
Feature filt.	0.15	0.11	0.29	sec
Features	19	21	26	
Total time	0.94	2.03	10.68	sec

Table 1. Vision sensor timing.

The main processing time is used for the stereoscopic scanning as would be expected. At the 160x120 resolution, the epipolar distance is six pixels, and as the scanning mask is three pixels wide there is an almost full usage of all pixels in the image. The result is available after about one second. The 320x240 resolution produces slightly better data, but takes 2 seconds to complete.

In most scenes, the sensor produce from 15 to 40 clusters, where each cluster represents a visible feature from *one* of the objects in the scene.

The delay in camera and interface limits the accuracy of image time stamping, and thereby the accuracy of the position and orientation of the camera, especially if the images was taken during a maneuver. This, in turn, limits the accuracy of the stereo extraction.

The method is rather insensitive to background structure, as can be seen in Figure 7, where the closed doors have been opened. The method extracts readily more complex shaped objects as the toy Jeep in Figure 8.

Isolation of features from reasonably sized objects — e.g. the table and the chair — seems adequate for a reasonably safe classification. The 3D extension of objects may be used to get further data

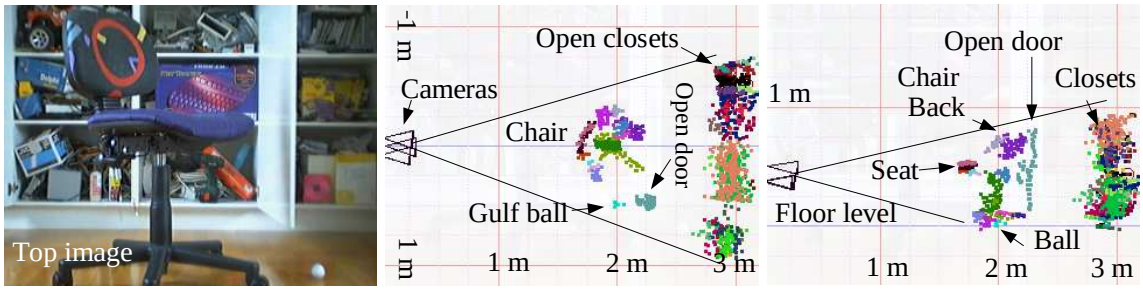

Fig. 7. The same chair and golf ball as in Figure 3, but with more complex background. The chair and ball is still extracted. Center image is top view and right image is side view.

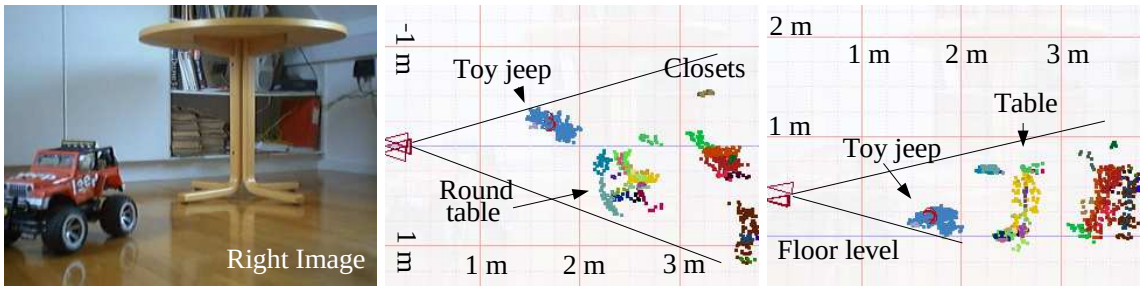

Fig. 8. The same method now used on a toy jeep and a round table. The objects are extracted to a level where the table should be recognizable as a table.

from the source images, e.g. additional color or texture information, which can be used to further distinguish objects and object classes. The additional color and texture extraction is not implemented yet; neither is the object classification.

7. CONCLUSION

The vision sensor described in this paper combines stereo vision methods used in e.g. land mapping from aerial photographs, and 3D filtering methods that in total ensures a robust extraction of all — reasonably sized — objects within sight of the robot. The processing time for the purposed method is not prohibitive, and will be even less so in the future.

The extracted objects seem to be adequate for classification of objects in classes like chairs, tables, walls and doors.

The process is designed for an in-door environment, but should also be applicable for out-door use, as there is no dependence on the shape of the seen features.

7.1 Next step

The next step is expected to be grouping of detected clusters to isolate non-transparent surfaces, e.g. walls, chair and table surfaces. The image color and texture of these surfaces can then be determined within the surface, to enhance the correlation and classification process.

REFERENCES

Babuška, Robert (1998). *Fuzzy Modeling for Control.* Kluwer Academic Publishers.

Kim, Dongsung and Ramakant Nevatia (1999). Symbolic navigation with a generic map. *Autonomous Robota* **6**, 69–88. Soongsil University, Korea and University of Southern California, USA.

Loaïza, H., J. Triboulet, S. Lelandais, F. Chavand and F. Artigue (1999). A multi-configuration stereoscopic vision system for domistic mobile robot localization. pp. 207–212.

Loaïza, Huberto, Jean Triboulet, Sylvie Lelandais and Christian Barat (2001). Matching segments in stereoscopic vision. *IEEE Instrumentation and Measurement Magazine* pp. 37–42.

Murray, Don and James J. Little (2000). Using real time stereo vision for mobile robot navigation. In: *Autonomous Robots.* Kluwer Academic Publishers, The Netherlands. University of British Columbia, Canada. pp. 161–171.

Se, Stephan, David Lowe and Jim Little (2001). Vision-based mobile robot localization and mapping using scale-invariant features. In: *IEEE International Conference on Robotics and Automation, Seoul, Korea.* IEEE. pp. 2051–2058.

Shah, Shishir and J. K. Aggarwal (1997). Mobile robot navigation and scene modelling using stereo fish-eye lens system. *Machine Vision and Application* **10**, 159–173.